Crawling Web berdasarkan Ontology

Eri Zuliarso[1] , Khabib Mustofa[2]
[1]Mahasiswa Pasca Sarjana Ilmu Komputer FMIPA UGM
[2]Staff Pengajar Program Pasca Sarjana Ilmu Komputer FMIPA UGM
ezuliarso@yahoo.com

Abstrak

Telah dikembangkan sebuah aplikasi Web Crawler untuk melakukan penjelajahan dan pengambilan halaman-halaman Web yang ada di Internet. Program crawl memanfaatkan WordNet dan ontology dari struktur Open Directory Project (ODP) untuk mencari relevansi suatu halaman web dengan kata kunci. Pentingnya suatu halaman web dihitung menggunakan rumus similaritas tekstual. Pengujian dilakukan untuk membandingkan *harvest-rate* crawling menggunakan WordNet dengan menggunakan ontology dari struktur ODP.

Kata kunci : *Crawler, WordNet, Ontology, ODP*

1. Pendahuluan

Pertumbuhan World Wide Web yang eksplosif membuat sukar menemukan informasi yang sesuai dengan keinginan pemakai. Terlalu banyak server dan halaman yang harus dilihat dan dilakukan secara on line tetap merupakan tugas yang mengkonsumsi waktu. Hal inilah yang disebut masalah penemuan sumberdaya internet (*internet resource discovery problem*) [2]

Web Crawler, juga sering dikenal sebagai Web Spider atau Web Robot adalah salah satu komponen penting dalam sebuah mesin pencari modern. Fungsi utama Web Crawler adalah untuk melakukan penjelajahan dan pengambilan halaman-halaman Web yang ada di Internet. Hasil pengumpulan situs Web selanjutnya akan diindeks oleh mesin pencari sehingga mempermudah pencarian informasi di Internet.

Mendesain sebuah crawler yang baik saat ini menemui banyak tantangan [7]. Secara eksternal, crawler harus mengatasi besarnya situs Web dan link jaringan. Secara internal, crawler harus mengatasi besarnya volume data. Sehubungan dengan terbatasnya sumber daya komputasi dan keterbatasan waktu, maka harus hati-hati memutuskan URL apa yang harus di scan dan bagaimana urutannya. Crawler tidak dapat mengunduh semua halaman web. Penting bagi crawler untuk memilih halaman dan mengunjungi halaman yang penting dulu dengan memprioritaskan URL yang penting tersebut dalam antrian. Crawler juga harus memutuskan berapa frekuensi untuk merevisi halaman yang pernah dilihat, untuk memberikan informasi ke client perubahan yang terjadi di Web.

Kebanyakan crawler tidak dapat mengunjungi setiap halaman web yang mungkin, dengan dua alasan utama :

- Client mempunyai kapasitas penyimpan yang terbatas, dan tidak dapat untuk mengindeks atau menganalisa semua halaman.
- Crawling memakan waktu, pada suatu waktu crawler mungkin perlu mulai mengunjungi halaman yang telah discan sebelumnya, untuk menguji perubahan.

Karena crawler hanya dapat mengunduh sebagian kecil dari halaman Web, maka crawler perlu secara hati-hati memutuskan halaman mana yang perlu diunduh. Dalam paper ini akan dibahas hasil eksperimen crawler berbasis similaritas berdasarkan Wordnet dan domain ontology. Metrik penting berbasis similaritas mengukur relevansi tiap halaman ke sebuah topic atau query yang diberikan oleh pemakai.

2. Dasar Web Crawler

Walaupun banyak aplikasi untuk Web crawler, pada intinya semuanya secara fundamental sama[4]. Berikut ini proses yang dilakukan Web crawler pada saat bekerja :

a. Mengunduh halaman Web.

b. Memparsing halaman yang didownload dan mengambil semua link.

c. Untuk setiap link yang diambil, ulangi proses.

Dalam langkah pertama, sebuah web crawler mengambil URL dan mengunduh halaman dari Internet berdasarkan URL yang diberikan. Seringkali halaman yang diunduh disimpan ke sebuah file atau ditempatkan di basisdata. Dengan menyimpan halaman web, maka crawler atau program yang lain dapat memanipulasi halaman itu untuk diindeks (dalam kasus mesin pencari) atau untuk pengarsipan untuk digunakan oleh pengarsip otomatis.

Tahap kedua, Web crawler memparsing keseluruhan halaman yang diunduh dan mengambil link-link ke halaman lain. Tiap link dalam halaman didefinisikan dengan sebuah penanda HTML yang serupa dengan yang ditunjukkan disini :

<A HREF="http://www.host.com/directory/file.html ">Link</A>

Setelah crawler mengambil link dari halaman, tiap link ditambahkan ke sebuah daftar untuk dicrawl.

Langkah ketiga dari Web crawling adalah mengulangi proses. Semua crawler bekerja dengan rekursif atau bentuk perulangan, tetapi ada dua cara berbeda untuk menanganinya. Link dapat dicrawl dalam cara depth-first atau breadth-first.

Depth-first crawling mengikuti tiap kemungkinan jalur sampai selesai sebelum mencoba jalur yang lain. Algoritma ini bekerja dengan menemukan link pertama pada halaman pertama. Kemudian mengcrawl halaman yang berasosiasi dengan link tersebut, menemukan link pertama pada halaman pertama dan begitu seterusnya sampai ujung dari jalur dicapai. Proses terus berlanjut sampai semua cabang dari link telah dikunjungi.

Breadth-first crawling menguji tiap link pada sebuah halaman sebelum memproses ke halaman berikutnya. Jadi, algoritma ini menelusuri tiap link pada halaman pertama dan kemudian menelusuri tiap link pada halaman pertama pada link pertama dan begitu seterusnya sampai tiap level pada link telah dikunjungi.

3. *Robot Protocol*

Crawling sebuah situs Web dapat membuat ketegangan yang besar pada sumberdaya server Web karena berulangkali permintaan dibuat kembali dan kembali. Biasanya, beberapa halaman didownload pada suatu waktu dari halaman Web, tidak beratus-ratus atau ribuan secara berurutan. Situs-situs web juga sering mempunyai area yang dibatasi sehingga crawler seharusnya tidak menelusuri. Untuk menampung pertimbangan ini, banyak situs Web mengadopsi *Robot protocol,* yang menetapkan petunjuk yang seharusnya diikuti oleh crawler. Seiring berjalannya waktu, protocol menjadi hukum tidak tertulis di Internet untuk crawler Web.

Robot protocol menspesifikasikan bahwa situs Web membatasi area tertentu atau halaman dari crawling disimpan dalam sebuah file yang diberi nama robots.txt yang ditempatkan pada root pada situs Web. Secara etika crawler akan mereferensi file robot dan menentukan bagian mana dari situs yang tidak diperbolehkan untuk ditelusuri. Area yang tidak diperbolehkan akan dilompati oleh crawler yang beretika.

4. Metrik yang penting

Tidak semua halaman web perlu mendapat perhatian yang sama untuk sebuah crawler. Sebagai contoh, jika crawler digunakan untuk membangun suatu basisdata yang khusus untuk topik tertentu, maka halaman yang merujuk ke topik tersebut lebih penting, dan seharusnya dikunjungi seawal mungkin.

Jika diberikan sebuah halaman web p, didefinisikan pentingnya suatu halaman *I(p)*, dalam salah satu cara berikut[7] .

a. ***Similaritas untuk mengendalikan query Q*** : Sebuah query *Q* mengendalikan proses

crawling, dan *I(p)* didefinisikan sebagai similaritas tekstual diantara *p* dan *Q*.

Untuk menghitung similaritas, dapat dilihat tiap dokumen halaman web sebagai vektor berdimensi *n* $(w_1,w_2,.....,w_n)$. Term w_i adalah vektor yang merepresentasikan kata ke *i* dalam pustaka. Jika w_i tidak muncul dalam dokumen, maka w_i nol. Jika muncul, w_i di set untuk merepresentasikan signifikansi dari kata. Salah satu cara yang umum untuk menghitung signifikansi w_i adalah dengan mengalikan banyaknya kemunculan kata *i* di dokumen dengan *inverse document frequency (idf)* dari kata ke-*i*. Faktor *idf* adalah satu dibagi dengan banyaknya kemunculan kata dalam keseluruhan koleksi.

b. ***Forward Link Count*** : Metrik ***IF(p)*** menghitung banyaknya link yang berasal dari *p*. Menggunakan metric ini, sebuah halaman dengan link keluar yang banyak sangat berharga, karena merupakan sebuah directory Web. Metrik ini dapat dihitung secara langsung dari *p*.

c. ***Backlink Count****:* Harga dari *IB(p)* adalah banyaknya gandengan (link) ke *p* yang muncul di seluruh Web. Secara intuitif, sebuah halaman *p* yang dilink oleh banyak halaman lebih penting daripada halaman yang jarang menjadi referensi. Tipe "penghitungan kutipan" telah banyak digunakan secara luas untuk mengevaluasi pengaruh dari makalah yang dipublikasikan.

d. ***PageRank.*** Metric IB(P) menetapkan semua link sama. Jadi, sebuah link dari homepage Yahoo dihitung sama dengan link dari beberapa homepage individu. Namun demikian, karena home page Yahoo lebih penting (memiliki perhitungan *IB* yang lebih tinggi), adalah masuk akal untuk memberikan harga link ke Yahoo lebih tinggi.

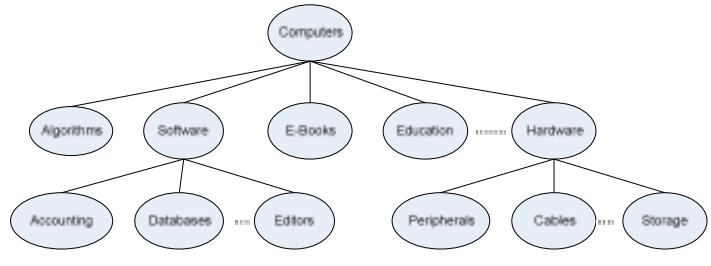
e. ***Location metric*** : ***IL(p)*** pentingnya halaman p sebagai fungsi lokasi, bukan dari isinya. Sebagai contoh, URL yang berakhir dengan ".com" dianggap lebih berguna daripada URL dengan akhiran yang lain.

5. Ontology Web

Istilah ontology berasal dari filsafat. Dalam konteks ini, ontology digunakan sebagai subbidang dari filsafat, yang mempelajari sifat alami dari keberadaan, cabang dari metafisik yang berkaitan dengan identifikasi, atau secara umum, jenis-jenis benda yang secara actual ada, dan bagaimana memaparkannya. Sebagai contoh, observasi yang dilakukan pada objek tertentu yang mengelompokkan menjadi kelas-kelas abstrak berdasarkan pada sifat-sifat bersama merupakan komitmen ontology secara tipe.

Namun demikian, dalam beberapa tahun belakangan, ontology menjadi kata yang diambil oleh ilmu komputer dan diberikan sebuah arti teknis khusus yang sedikit berbeda dari aslinya.

Menurut definisi T.R. Gruber's, yang kemudian diperbaiki R. Studer[6]: Sebuah ontology adalah spesifikasi yang eksplisit dan formal dari sebuah konseptualisasi. Secara umum, sebuah ontology memaparkan secara formal sebuah domain topik pembicaraan. Sebuah ontology terdiri dari sebuah daftar istilah terbatas dan hubungan diantara istilah-istilah ini. *Istilah* menandakan pentingnya *konsep (kelas dari objek)* dari suatu domain. Sebagai contoh, dalam dunia komputer, ada algorithm, software, e-books dan hardware adalah konsep yang penting. Hubungan (*relationship*) mencakup hirarki dari kelas-kelas. Sebuah hirarki menspesifikasikan sebuah kelas menjadi subkelas C_ yang lain jika setiap objek dalam C juga termasuk dalam C_. Sebagai contoh, diperlihatkan hirarki sebuah domain computers.



Gambar 1. Hirarki domain Computers
(diambil dari http://www.dmoz.org)

Ontology sangat berdaya guna untuk organisasi dan navigasi situs web. Banyak situs

web saat ini menyajikan sisi sebelah kiri halaman web, daftar tingkat tertinggi dari hirarki konsep suatu istilah. Pemakai mungkin mengklik salah satu pilihan untuk memperluas subkategori. Ontology juga meningkatkan keakuratan pencarian Web. Mesin pencari dapat mencari halaman-halaman yang menunjuk ke konsep yang tepat dalam sebuah ontology. Mesin pencari web juga dapat mengeksploitasi informasi generalisasi atau spesialisasi. Jika suatu query gagal untuk menemukan dokumen yang relevan, mesin pencari dapat menyarankan pemakai sebuah query yang lebih general. Atau untuk mencegah terlalu banyak jawaban yang diberikan, maka mesin pencari dapat menyarankan pencarian yang khusus (spesialisasi).

6. WordNet

WordNet adalah sebuah leksikal database yang dibangun oleh Cognitive Science Lab. Princeton University. WordNet adalah sebuah basisdata linguistik yang dibangun dengan *synsets*-istilah-istilah yang dikelompokkan menjadi himpunan-himpunan yang ekuivalen secara semantic, masing-masing istilah dimasukkan ke kategori leksikal (kata benda, kata kerja, kata keterangan, kata sifat) [3]. Tiap *synset* menyatakan sebuah pengertian (*sense)* tertentu dari sebuah kata bahasa Inggris dan biasanya dinyatakan sebagai sebuah kombinasi unik dari kata-kata yang bersinonim. Sebuah kata yang memiliki lebih dari satu pengertian disebut *polysemous*; dua kata yang memiliki paling tidak satu pengertian bersama disebut sinonim.

Hubungan semantic dalam WordNet mencakup :

- *Sinonim* adalah hubungan dasar dalam WordNet, karena WordNet menggunakan sekumpulan sinonim (*synset*) untuk menyatakan pengertian kata.
- *Antonim* (lawan kata) juga merupakan hubungan semantic diantara bentuk kata, khususnya untuk mengorganisir arti kata sifat dan kata keterangan.
- *Hyponymy* (nama sub-kelas) dan kebalikannya, *hypernymy* (nama super-kelas), digunakan untuk menggambarkan hubungan semantic terorganisasi kata benda menjadi struktur hirarki.
- *Meronymy* (nama bagian) dan kebalikannya, *holonymy* (nama keseluruhan).

Sebuah synset (synonym set) adalah himpunan kata-kata dengan kesamaan part-of-speech yang dapat dipertukarkan dalam konteks tertentu. Sebagai contoh, {car; auto; automobile; machine; motorcar} membentuk synset karena kata-kata tersebut dapat digunakan untuk menunjuk ke konsep yang sama. Sebuah synset sering dipaparkan lebih lanjut dengan sebuah penjelasan : "4-wheeled; usually propelled by an internal combustion engine". Akhirnya, synset dapat dihubungkan dengan yang lain dengan hubungan semantic seperti hyponymy (diantara konsep khusus dan yang lebih umum), meronymy (diantara bagian dan keseluruhan). Seperti diilustrasikan pada gambar 2

Gambar 2. Contoh hubungan kata dalam WordNet

Dalam contoh ini, diambil dari WordNet 1.5, synset {car; auto; automobile; machine; motorcar} berhubungan dengan :

- Sebuah konsep yang lebih umum atau synset hyperonym : {motor vehicle; automotoive vehicle},
- Konsep yang lebih khusus atau synset hyponym; ebagai contoh {cruiser; squad car; patrol car; police car; prowl car} dan {cab; taxi; hack; taxicab},
- Bagian terdiri dari : sebagai contoh {bumper};{car door},{car mirror} dan {car window}

Tiap-tiap synset ini sekali lagi berelasi dengan synset yang lain sebagaimana ilustrasi untuk {motor vehicle;automotive vehicle} yang

berelasi dengan {vehicle}, dan {car door} yang berelasi dengan bagian yang lain: {hinge; flexible joint},{armrest},{doorlock}. Dengan alat seperti ini dan relasi semantic dan konseptual yang lain, semua arti kata dalam suatu bahasa dapat saling terhubung, membuat suatu jaringan besar atau wordnet. Beberapa wordnet dapat digunakan untuk membuat inferensi secara semantic (*what things can be used as vehicles*), mencari ekspresi alternatif atau wordings (*what words can refer to vehicles*), atau untuk secara sederhana memperluas kata ke himpunan yang secara semantic berelasi atau kata yang artinya dekat, sebagai contoh dalam information retrieval.

7. Definisi Masalah

Tujuan untuk mendesain crawler adalah jika mungkin mengunjungi halaman dengan *I(p)* tinggi sebelum halaman dengan *I(p)* yang lebih rendah untuk beberapa definisi *I(p)*. Adapun model pengujian suatu crawler adalah sebagai berikut :

- Dalam melakukan penelusuran, crawler menggunakan model **Crawl dan Stop**. Dalam model ini, crawler C mulai pada halaman awal p_o dan berhenti setelah mengunjungi *K* halaman. Pada titik ini sebuah crawler yang sempurna harus sudah mengunjungi halaman $r_1,...,r_K$ dimana r_1 adalah halaman dengan harga terpenting, *r2* yang penting berikutnya, begitu berikutnya. Halaman $r_1,...,r_K$ disebut halaman "*hot*". *K* halaman yang akan dikunjungi oleh crawler pada penelitian ini hanya akan memuat *M* halaman dengan rangking lebih tinggi atau sama dengan $I(r_K)$. Kinerja crawler *C* didefinisikan sebagai $P_{CS}(C)= M/K$.. Kinerja dari crawler yang ideal sudah tentu 1. Dalam [9] dikenal metrik ***harvest-rate :***

$$hr = \frac{\#r}{\#p}, hr \in [0,1]$$

Harvest-rate merepresentasikan pecahan halaman Web yang ditelusuri yang memenuhi target ***#r*** dalam keseluruhan halaman ***#p*** yang diperoleh. Rasio harvest-rate ini harus tinggi, jika tidak crawler akan banyak meluangkan waktu untuk mengeliminasi halaman yang tidak relevan.

- **Crawl and Stop with Threshold**. Diasumsikan bahwa crawler mengunjungi *K* halaman. Namun demikian, crawler diberi target penting *G*, dan sembarang halaman dengan $I(p) \geq G$ termasuk sebagai halaman "hot". Diasumsikan bahwa total banyaknya halaman yang hot adalah *H*. Kinerja dari crawler, $P_{ST}(C)$, perbandingan halaman hot *H* dengan *K* halaman yang dikunjungi ketika crawl telah berhenti. Jika $K < H$, maka crawler yang ideal memiliki kinerja K / H. Jika $K \geq H$, maka crawler yang ideal mempunyai kinerja yang sempurna 1.

Dalam eksperimen berikut ini dicoba program crawl dengan berbasis kata kunci dengan memanfaatkan Wordnet dan ontology dari struktur ODP. Untuk menghitung relevansi suatu halaman web dengan kata kunci dihitung menggunakan rumus similaritas tekstual. Karena tidak memungkinkan untuk mengumpulkan seluruh halaman web, maka dibuat asumsi jika dalam suatu halaman web terdapat 10 kata yang sesuai dengan kata kunci, maka halaman tersebut dianggap relevan dengan query pemakai [7].

Sebagai perbandingan dicoba juga crawler dengan similaritas tekstual murni, jadi akan diuji berdasarkan pencocokan suatu kata yang dimasukkan dengan suatu kata di halaman web.

Pada crawling ini, akan dibandingkan tiga teknik penelusuran:

1. Menggunakan synset dari suatu kata dalam WordNet.
2. Menggunakan relasi hiponim dari suatu kata dalam WordNet.
3. Menggunakan struktur ontology dari suatu kata berdasarkan struktur hirarki http://www.dmoz.org.

Dalam WordNet, kata "*computer*" mempunyai synset :

computer, computing machine, computing device, data processor, electronic computer, information processing system

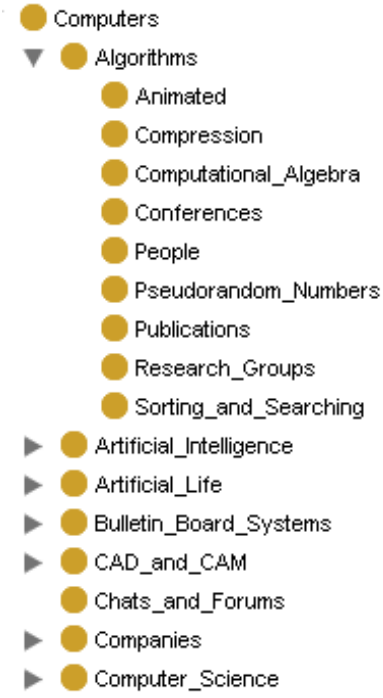
Sedangkan kata yang merupakan hiponim dari kata "computer" adalah :

computer, computing machine, computing device, data processor, electronic computer,

information processing system -- (a machine for performing calculations automatically)

- ⇨ analog computer, analogue computer -- (a computer that represents information by variable quantities (e.g., positions or voltages))
- ⇨ digital computer -- (a computer that represents information by numerical (binary) digits)
- ⇨ home computer -- (a computer intended for use in the home)
- ⇨ node, client, guest -- ((computer science) any computer that is hooked up to a computer network)
- ⇨ number cruncher -- (a computer capable of performing a large number of mathematical operations per second)
- ⇨ pari-mutuel machine, totalizer, totaliser, totalizator, totalisator -- (computer that registers bets and divides the total amount bet among those who won)
- ⇨ predictor -- (a computer for controlling antiaircraft fire that computes the position of an aircraft at the instant of a shell's arrival)
- ⇨ server, host -- ((computer science) a computer that provides client stations with access to files and printers as shared resources to a computer network)
- ⇨ turing machine -- (a hypothetical computer with an infinitely long memory tape)
- ⇨ web site, website, internet site, site -- (a computer connected to the internet that maintains a series of web pages on the World Wide Web; "the Israeli web site was damaged by hostile hackers")

Sebagian struktur taksonomi ontologi dari kata “Computers” pada http://www.dmoz.org adalah sebagai berikut :



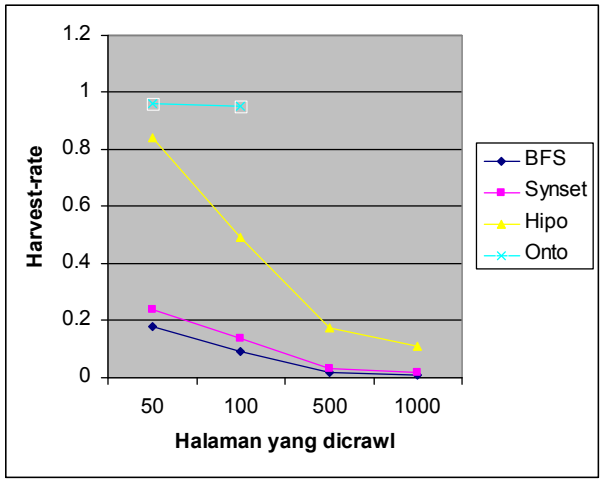
Gambar 3. Struktur taksonomi ontology dari kata ‘computers’

8. Hasil eksperimen

Dalam eksperimen ini, digunakan definisi *IS(p)* : sebuah halaman dipertimbangkan sebagai hot jika halaman itu memuat kata “computer” lebih dari 10 . Sedangkan untuk halaman URL digunakan halaman http://www.amazon.com.

Pada saat eksperimen pertama digunakan synset dalam WordNet untuk suatu kata. Jika ada kecocokan untuk tiap synset akan dihitung. Pada eksperimen kedua, digunakan hiponim dalam WordNet untuk suatu kata, maka jika ada kecocokan untuk tiap kata yang menjadi hiponim akan dihitung. Dalam eksperimen ketiga, digunakan struktur ontology suatu kata berdasar struktur hirarki Open Directory Project (http://www.dmoz.org).

Adapun perbandingan kinerja crawl keseluruhan ditampilkan dalam grafik di bawah ini.



Gambar 4. Grafik Hasil Eksperimen

Sumbu X dalam grafik menyatakan banyaknya halaman yang dicrawl. Sedangkan sumbu Y menyatakan **harvest-rate**. Sebagai pembanding digunakan Metode Breadth-first untuk membandingkan dengan 3 cara yang lain. Dari hasil percobaan, didapatkan bahwa menggunakan BFS, banyaknya halaman yang sesuai tetap. Sehingga semakin banyak halaman yang dicrawl, kinerja akan semakin menurun.

Dengan menggunakan synset dari WordNet, membuka kesempatan pencocokan lebih dari satu kata. Sehingga ratio halaman yang relevan lebih baik dari pada BFS. Namun demikian, karena sinonim suatu kata terbatas, maka semakin banyak halaman yang di crawl, akan semakin banyak yang tidak relevan.

Pada eksperimen menggunakan kata yang merupakan suatu hiponim (subkelas) dari suatu kata dalam WordNet memberikan hasil yang bagus pada pengujian dengan halaman yang dicrawl sedikit. Relasi hiponim memberikan variasi makna yang lebih banyak dibandingkan dengan relasi sinonim.

Hasil terbaik diperoleh dengan menggunakan struktur ontology. Hal ini dikarenakan, struktur ontology mempunyai hirarki pengetahuan yang lebih luas dibandingkan dengan hubungan sinonim ataupun hiponim. Sehingga apabila dilakukan crawling dengan jumlah halaman yang lebih banyak, kinerja harvest-rate masih dapat dipertahankan dengan harga yang tinggi.

9. Kesimpulan

Dalam paper ini dipaparkan eksperimen untuk melakukan crawling dan melakukan pemilihan suatu halaman web, apakah halaman web tersebut penting atau tidak. Penting atau tidaknya suatu halaman web dihitung menggunakan metric berbasis similaritas dengan membandingkan berdasarkan synset WordNet, hiponim dalam WordNet dan domain ontology. Dari hasil eksperimen didapat crawling menggunakan ontology menghasilkan harvest-rate yang lebih baik dibandingkan dengan menggunakan synset dan hiponim dalam WordNet.

Dalam eksperimen ini hasil perhitungan metric similaritas belum digunakan untuk menentukan urutan link yang akan di crawl.

Daftar Pustaka

[1] Antoniou, G.,F.v. Harmelen, 2008, *A Semantic Web Primer 2nd Ed.*, MIT Press

[2] Budi Yuwono, Savio L. Y. Lam, Jerry H. Ying, Dik L. Lee, 1996, *A World Wide Web Resource Discovery System*, in Proceedings of ICDE

[3] G.A. Miller, 1995, "WORDNET : A Lexical Database for English", *Communication ACM, Vol 2, No. 11, Nov. 1995, pp 39-41.*

[4] G. Pant, P. Srinivasan, F. Menczer, "*Crawling the Web",*

[5] Gomez-Perez, A., O. Corcho, *Ontology Languages for the Semantic Web, IEEE Intelligent, January / February, 2002*

[6] Gruber, T., 1998, *What is an Ontology?, http://www-ksl.stanford.edu/kst/what-is-an-ontology.html*

[7] Junghoo Cho, Hector Garcia-Molina, and Lawrence Page, **"**Efficient crawling through URL ordering**",** *In Proceedings of the Seventh International World Wide Web Conference*, pages 161--172, April 1998

[8] Marc Ehrig and Alexander Maedche, 2003, "*Ontology-Focused Crawling of Web Documents", In Proceedings of*

Symposium on Applied Computing,Florida, USA

[9] Nicola Guarino, Claudio Masolo, Guido Vetere, 1999, "OntoSeek: Content-Based Access to the Web", IEEE Intelligents System

[10] Open Directory Project, http://www.dmoz.org

[11] P Srinivasan, F. Menczer, G. Pant, 2004, "*A General Evaluation Framework for Topical Crawlers", Kluwer Academic Publishers*

[12] S. Chakrabarti, M. van den Berg, and B. Dom, "*Focused crawling: a new approach to topic-specific Web resource discovery", Computer Networks (Amsterdam, Netherlands),* vol 31, pp. 1623-1640, 1999. Available : citeseer.ist.psu.edu/chakrabarti99focused.html

[13] Sergey Brin and Lawrence Page, 1998**,** **"**The anatomy of a large-scale hypertextual Web search engine**",** In *Proceedings of the Seventh International World Wide Web Conference*, pages 107--117, April 1998

[14] S. Ganesh, M. Jayaraj, V. Kalyan Srinivasal Murthy, G. Aghila, 2004, "*Ontology-based Web Crawler", In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), IEEE*